ISSN 1561-5359. Штучний інтелект, 2017, № 2

УДК 004.37

П.Я. Пукач, Х.Р. Шаховська
Національний університет «Львівська політехніка», Україна
вул. С. Бандери, 12, м. Львів, 79013

АЛГОРИТМ ФОРМУВАННЯ ВІДПОВІДІ ЧАТ-БОТА

P.Ya. Pukach, Kh.R. Shakhovska
Lviv Polytechnic National University, Ukraine
12, S. Bandery str., Lviv, 79013

MAKING ANSWER ALGORITHM FOR CHAT-BOT

У статті досліджено недоліки існуючих чат-ботів та розглянуто методи їх покращення. Зокрема, запропоновано використання алгоритмів Рабіна-Карпа та Кнута-Пратта для формування відповіді користувачеві та показано їх ефективність.

Ключові слова: чат-бот, хешування, префікс-функція.

This paper explores pro and cons of existing chat-bots and investigate methods of their improvement. Particularly, is proposed usage of Rabin-Karp and Knut-Pratt algorithms for making answer to user and is demonstrated their effectivity.

Keywords: chat-bot, hashing, prefix-function.

Вступ

Робот або бот [1], а також інтернет-бот, www-бот тощо − спеціальна програма, що виконує автоматично і/або за заданим розкладом які-небудь дії через ті ж інтерфейси, що й звичайний користувач. Під час обговорення комп'ютерних програм термін уживається в основному в застосуванні до Інтернету. Зазвичай боти призначаються для виконання роботи, одноманітної й повторюваної, з максимально можливою швидкістю (очевидно, набагато вищою за можливості людини). Людина, що обслуговує сервери, може помістити на сервері файл robots.txt, що містить обмеження, яким зобов'язані підпорядковуватися боти.

Чат-бот може видати досить адекватну відповідь на питання. Такі боти часто застосовуються для повідомлення прогнозу погоди, результатів спортивних змагань, курсів валют, біржових котирувань тощо. Вони знаходять застосування, наприклад, у системі SmarterChild в AOL Instant Messenger і MSN messenger. Також чат-боти використовуються інтернет-магазинами для спілкування з користувачами та промоції продукції. Інша сфера застосування – соціальні мережі, де такі програми, фактично, можуть задавати «тон» бесіді.

Проте недоліком бота є те, що його відповідь має неузгоджені відмінок, рід тощо. Часто боти можуть відповідати на запитання із заданої ключовими словами тематики.

Метою роботи є розроблення алгоритму аналізу та розбору тексту користувача для автоматичного формування відповіді чат-бота з врахуванням тематики переписки та морфології тексту. Робота алгоритму базуватиметься на префікс-функції та хеш-функції. Також буде здійснено порівняння розробленого алгоритму з існуючими.

Аналіз літературних джерел

За визначенням [1] інтелект чат-бота – це те, що допомагає йому керувати будь-яким сценарієм бесіди.

Найпоширенішими є боти, які за допомогою кнопкового інтерфейсу дають відповідь на запитання. Проте предметом наших досліджень є боти, які «розуміють» природну мову (з nature language interface). Найкращі онлайн-чати на основі штучного інтелекту (ШІ) − Mitsuku, Rose, Poncho, Right Click, Insomno Bot, Dr. AI та Melody.

1. Міцуку [2].

Є одним з кращих зі ШІ, нинішній лауреат премії Лоєбнера. Премія Лоєбнера – це щорічний конкурс методів штучного інтелекту, у якому визначається інтелект, найбільше подібний до людського. Формат конкурсу –стандартний тест Тьюрінга. Цей бот може поговорити про що-небудь, на відміну від інших, зроблених для конкретного завдання.

2. Роза [3].

Чат-бот, який отримав премію Лоєбнера в 2014 і 2015 роках.

3. RightClick [4].

Запускає програму, що створює веб-сайти. Він ставить загальні питання під час бесіди: «Яка сфера Ваших інтересів?» та «Чому ви хочете створити веб-сайт?». На основі аналізу отриманих відповідей, чат-бот створює індивідуальні шаблони. Доволі адекватно реагує на відповіді, що не стосуються тематики створення сайтів.

4. Пончо [5].

Пончо – метеорологічний фахівець. Він надсилає сповіщення щонайменше двічі на день за згодою користувача та досить розумний, щоб відповідати на такі запитання, як «Я повинен взяти парасольку сьогодні?»

5. Інсономічний бот [6].

Інсономічний бот призначений для нічних сов. Як випливає з назви, це стосується всіх людей, які мають проблеми зі сном. Цей бот має змогу підтримувати розмову на будь-яку тему.

6. Д-р A.I. [7].

Бот запитує про симптоми, параметри тіла та історію хвороби, потім складає список найбільш і найменш вірогідних причин симптомів і сортує їх за порядком серйозності.

7. Мелодія від Baidu [8].

Цей додаток збирає медичну інформацію про людей, а потім передає її лікарям у формі, що полегшує її використання для діагностичних цілей.

Більша частина будь-якого чат-бота – це розмова з її користувачем. Отже, основну увагу у алгоритмі бота необхідно приділяти розробці бесіди в чаті. Для англійської мови доволі легко можна збудувати речення, оскільки визначення особи та множини/однини доволі легко зробити. Для таких мов як німецька важливим у правильному морфологічному формуванні відповіді є порядок слідування слів у реченні, що також може бути доволі формалізованим. Якщо йде мова про створення ботів для слов’янських мов (наприклад, української), то основною проблемою тут є не строгість розміщення слів у реченні, наявність різних закінчень для осіб та однини/множини, а також велика синонімічна база.

Постановка задачі

Задачею, яка вирішується у статті, є визначення статі співрозмовника для формування зв’язних повідомлень. Для цього ми створимо базу даних з ключовими словами (слова, на які повинен реагувати бот), відповідями на них та особовими закінченнями дієслів.

Префікс-функція [9, 10] рядка π (S, i) визначає довжину найбільшого префікса рядка S [1..i], який не збігається з цим рядком і одночасно є її суфіксом. Простіше кажучи, це довжина найдовшого початку рядка, що є також і його кінцем. Для рядка S зручно представляти префікс-функцію у вигляді вектора довжиною |S| -1. Можна розглядати префікс-функцію довжини |S|, поклавши π (S, 1) = 0. Приклад префікс-функції для рядка «abcdabcabcdabcdab»:

S[i]	a	b	c	d	a	b	c	a	b	c	d	a	b	c	d	a	b
π(S,i)	0	0	0	0	1	2	3	1	2	3	4	5	6	7	4	5	6

Хешування [11] (англ. Hashing) – перетворення масиву вхідних даних довільної довжини (масиву рядків) у (вихідний) бітовий рядок фіксованої довжини, що виконується певним алгоритмом. Функція, що реалізує алгоритм і виконує перетворення, називається хеш-функцією або функцією згортки. Вихідні дані називаються вхідним масивом, ключем або повідомленням. Результат перетворення (вихідні дані) називається хешем, хеш-кодом, хеш-сумою, зведенням повідомлення.

Хешування застосовується у таких випадках:

– для побудови асоціативних масивів;
– для пошуку дублікатів у серіях наборів даних;
– для побудови унікальних ідентифікаторів у наборах даних;
– для обчислення контрольних сум від даних (сигналу) для подальшого виявлення в них помилок (які виникли випадково або внесені навмисно), що виникають при зберіганні і / або передачі даних;
– для збереження паролів у системах захисту у вигляді хеш-коду (для відновлення пароля для хеш-коду потрібна функція, яка є зворотною щодо використаної хеш-функції);
– для вироблення електронного підпису (на практиці часто підписують не саме повідомлення, а його «хеш-образ») та ін.

Загалом (згідно з принципом Діріхле) немає однозначної відповідності між вихідними (вхідними) даними і хеш-кодом (вихідними даними). Значення (вихідні дані), що повертаються хеш-функцією, менш різноманітні, ніж значення вхідного масиву (вхідні дані). Випадок, при якому хеш-функція перетворює кілька різних повідомлень в однакові зведення, називається «колізією». Імовірність виникнення колізій використовується для оцінки якості хеш-функцій.

Розроблення алгоритму чат-бота

Алгоритм чат-бота полягатиме у пошуку ключового слова (слів) бесіди та формування особового закінчення дієслова.

Пошук ключового слова здійснимо за допомогою хешування. Для цього використаємо формулу розрахунку:

$$h(S) = S[0] + S[1] * P + S[2] * P^2 + S[3] * P^3 + ... + S[N] * P^N,$$

де P – просте число. Виберемо таке P, яке приблизно дорівнює кількості символів у вхідному алфавіті. Наприклад, якщо рядки складаються тільки з маленьких українських літер, то хорошим вибором буде P = 37. Використаємо алгоритм Рабіна-Карпа для пошуку підрядка в рядку за O (N).

Нехай задано текст користувача T і рядок S у ньому, що складаються з маленьких кириличних літер. Потрібно знайти всі входження рядка S у текст T за час $O(|S| + |T|)$. Алгоритм пошуку ключового слова складається з таких кроків:

1. Порахуємо хеш для рядка S.
2. Порахуємо значення хеш для всіх префіксів рядки T.
3. Переберемо всі підрядки T довжини |S|. Кожен з них можна порівняти з іншими рядками довжини |S| за час O (1).

Після того, як знайдено ключове слово, потрібно визначити особове закінчення дієслова. З цією метою намагаємося знайти дієслово як слово що знаходиться перед

або за знайденим ключовим словом. Перевірку закінчення реалізуємо за допомогою префікс-функції: за алгоритмом Кнутта-Пратта, який не містить явних порівнянь рядків і виконується за O (n) дій. Наведемо схему алгоритму:

1. Рахуватимемо значення префікс-функції $\pi[i]$ для $i \in [1..n-1]$ ($\pi[0] = 0$).
2. Для підрахунку поточного значення $\pi[i]$ використаємо змінну j, що позначає довжину поточного розглянутого зразка. Спочатку $j = \pi[i-1]$.
3. Тестуємо зразок довжини j, для чого порівнюємо символи s [j] і s [i]. Якщо вони однакові, то вважаємо $\pi[i]=j+1$, $i := i + 1$. Якщо ж символи відрізняються, то зменшуємо довжину j, вважаючи її рівною $\pi[j-1]$, і повторюємо цей крок алгоритму спочатку.
4. Якщо ми дійшли до довжини $j = 0$ і так і не знайшли однакових символів, то зупиняємо процес перебору зразків і вважаємо $\pi[i]=0$, $i := i + 1$.

Загальний алгоритм формування відповіді чат-ботом подано на рис. 1.



Рис.1. Алгоритм формування відповіді чат-ботом

Порівняння з існуючими методами

У першу чергу, порівняємо розроблений алгоритм з існуючими алгоритмами пошуку ключового слова. Для пошуку ключових слів використовують такі алгоритми:

1) Алгоритм прямого пошуку [12].

Ідея алгоритму:

1) I = 1,

2) порівняти I-й символ масиву T з першим символом масиву W,

3) якщо символи однакові, то порівняти наступні символи і так далі,

4) якщо символи різні, то I: = I + 1 і перехід на пункт 2.

Умова закінчення алгоритму:

1) M знайдених підряд порівнянь вдалі,

2) I + M> N, тобто слово, не знайдено.

Нехай масив T → {AAA ... .AAAB}, довжина │*T*│ – кількість усіх повідомлень, зразок W → {A ... .AB}, довжина │W│ = *S*. Очевидно, що для виявлення співпадіння в кінці рядка потрібно здійснити близько $S * T$ порівнянь, тобто O (S *T).

Недоліки алгоритму:

1) висока складність – O (S * T), у гіршому випадку – O ((T-S + 1) * T);
2) після виявлення під час порівняння різних символів, перегляд завжди починається з першого символу зразка і тому може включати символи T, які раніше вже були видимими (очевидно, що в он-лайн режимі таке порівняння не може бути реалізоване);
3) інформація про текст T, що отримується під час перевірки зміщення S, ніяк не використовується для перевірки наступних зміщень.

2) КМП-пошук [12].

Ідея КМП-пошуку полягає в тому, що під час кожної знайденої розбіжності двох символів тексту здійснюється зсув на довжину, яка дорівнює кількості символів, які співпали.

Особливості КМП-пошуку:

1) Потрібно близько (S + T) порівнянь символів для отримання результату.
2) Схема КМП-пошуку дає виграш тільки тоді, коли невдачі передувала певна кількість співпадінь. Лише у цьому випадку образ зсувається більше, ніж на одиницю. Тому обчислювальна складність цього алгоритму незначно відрізняється від обчислювальної складності прямого пошуку.

3) LSA [13].

Латентно-семантичний аналіз (LSA) призначений для визначення тематики тексту. Такий метод корисний для побудови чат-ботів, які не спеціалізуються на якійсь певній тематиці. Він складається з кроків:

1. Видалення стоп-слів, стемінг або лематизації слів у документах.
2. Видалення слів, що зустрічаються в єдиному екземплярі.
3. Побудова бінарної матриці: слово-документ SVD (можливо також частота входження).
4. Сингулярний розклад SVD = U * V * W^T, де U та W – ортогональні матриці, а V – діагональна, діагональні елементи впорядковані за спаданням.
5. Пошук рядків матриці U і стовпців W, які відповідають найбільшим сингулярним числам.

Недоліком LSA є припущення про те, що карта слів у документах не є нормально розподілена. Цей алгоритм добре піддається розпаралеленню, але, на жаль, для чат-бота це є неможливим. Складність алгоритму визначається $O(n^2 k^3)$, де n – кількість слів, k – розмірність матриці семантичного простору (для малих текстів лежить у межах від 50 до 350).

4) VSM зі tf-idf схемою.

Vector Space Model (VSM) зі tf-idf [14] схемою аналізує такі основні елементи:

- колекцію документів (переписка з користувачем), кожен з яких представлений у вигляді вектора;
- текстовий запит (відповідь користувача), також представлений у вигляді вектора.

Ми визначаємо K документів колекції з найвищим значенням векторного простору на запиті q. Як правило, документи впорядковуються за зменшенням цього значення. Складність алгоритму визначається як O(|Q|·|S| |T|), де Q – кількість термінів, |S| – кількість слів у відповіді користувача, |T| – кількість усіх повідомлень.

Зведене порівняння алгоритмів подано в таблиці 1.

Як бачимо, розроблений алгоритм не домінує тільки у КМП-пошуку в тому випадку, коли відразу знаходимо необхідний зразок.

Таблиця 1. Порівняння алгоритмів

Алгоритм	*Складність*
Розроблений алгоритм	$O(\|S\| + \|T\|)$
Прямий пошук рядка	$O((T\text{-}S + 1) * T)$
КМП-пошук	від $O(\|S\| + \|T\|)$ до $O((T\text{-}S + 1) * T)$
LSA	$O(n^2 k^3)$, n=$\|S\|\ \|T\|$
VSM зі tf-idf схемою	$O(\|Q\|\|S\|\|T\|)$

Висновки

Проаналізувавши існуючі методи, бачимо, що складність розробленого алгоритму пошуку ключових слів є меншою. Використання префікс-функції для формування відповіді бота дає змогу працювати з кириличними текстами. Запропонований алгоритм покращує роботу чат-ботів, а саме – визначає стать співрозмовника. Це наближує функціонування бота до рівня розмови людини.

Література

1. Shevat A. (2017). Designing bots: Creating conversational experiences (First ed.). Sebastopol, CA: O'Reilly Media. ISBN 9781491974827. OCLC 962125282
2. Міцуку // [Електр. Ресурс]. – Режим доступу: http://www.mitsuku.com/
3. Rose// [Електр. Ресурс]. – Режим доступу: https://www.robeco.nl/service-contact/index.jsp
4. Right click// [Електр. Ресурс]. – Режим доступу: https://rightclick.io/#/
5. Пончо // [Електр. Ресурс]. – Режим доступу: https://poncho.is/
6. Insomnobot// [Електр. Ресурс]. – Режим доступу: http://insomnobot3000.com/
7. Dr.A.I.// [Електр. Ресурс]. – Режим доступу: https://www.healthtap.com/login?redirect_to=/symptoms
8. Baidu Melody's// [Електр. Ресурс]. – Режим доступу: http://research.baidu.com/baidus-melody-ai-powered-conversational-bot-doctors-patients/
9. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ = Introduction to Algorithms / Под ред. И. В. Красикова. — 2-е изд. — М.: Вильямс, 2005. — 1296 с. — ISBN 5-8459-0857-4.
10. Knuth D., Morris J.H., Pratt Jr.V. (1977). «Fast pattern matching in strings». SIAM Journal on Computing 6 (2): 323-350. DOI:10.1137/0206024.
11. Кнут Д. Искусство программирования. Том 3. Сортировка и поиск = The Art of Computer Programming, vol.3. Sorting and Searching. — 2-е издание. — М.: «Вильямс», 2007. — С. 824. — ISBN 0-201-89685-0.
12. Урвачева В.А. «Обзор методов информационного поиска». Вестник Таганрогского института имени АП Чехова 1 (2016).
13. Landauer T.K. Latent semantic analysis. John Wiley & Sons, Ltd, 2006.
14. Aizawa A. «An information-theoretic perspective of tf–idf measures». Information Processing & Management 39.1 (2003): 45-65.

Literatura

1. Shevat A. (2017). Designing bots: Creating conversational experiences (First ed.). Sebastopol, CA: O'Reilly Media. ISBN 9781491974827. OCLC 962125282
2. Mitsuku // [Elektr. Resurs]. – Rezhym dostupu: http://www.mitsuku.com/
3. Rose // [Elektr. Resurs]. – Rezhym dostupu: https://www.robeco.nl/service-contact/index.jsp
4. Right click// [Elektr. Resurs]. – Rezhym dostupu: https://rightclick.io/#/
5. Poncho // [Elektr. Resurs]. – Rezhym dostupu: https://poncho.is/
6. Insomnobot// [Elektr. Resurs]. – Rezhym dostupu: http://insomnobot3000.com/
7. Dr.A.I// [Elektr. Resurs]. – Rezhym dostupu: https://www.healthtap.com/login?redirect_to=/symptoms
8. Baidu Melody's// [Elektr. Resurs]. – Rezhym dostupu: http://research.baidu.com/baidus-melody-ai-powered-conversational-bot-doctors-patients/

9. Kormen T., Leyzerson Ch., Ryvest R., Shtayn K. Alhorytm: postroenye y analyz = Introduction to Algorithms / Pod red. Y. V. Krasykova. ─ 2-e yzd. ─ M.: Vyl'yams, 2005. ─ 1296 s. ─ ISBN 5-8459-0857-4.
10. Knuth D., Morris J.H., Pratt Jr.V. (1977). «Fast pattern matching in strings». SIAM Journal on Computing 6 (2): 323–350. DOI:10.1137/0206024.
11. Knut D. Yskusstvo prohrammyrovanyya. Tom 3. Sortyrovka y poysk = The Art of Computer Programming, vol.3. Sorting and Searching. ─ 2-e yzdanye. ─ M.: «Vyl'yams», 2007. ─ S. 824. ─ ISBN 0-201-89685-0.
12. Urvacheva V.A. «Obzor metodov ynformatsyonnoho poyska». Vestnyk Tahanrohskoho ynstytuta ymeny AP Chekhova 1 (2016).
13. Landauer T.K. Latent semantic analysis. John Wiley & Sons, Ltd, 2006.
14. Aizawa A. «An information-theoretic perspective of tf–idf measures». Information Processing & Management 39.1 (2003): 45-65.

RESUME

P.Ya. Pukach, Kh.R. Shakhovska

Making answer algorithm for chat-bot

A robot or bot, as well as an internet bot, a www-bot, etc., is a special program that performs automatically and / or on a given schedule of any action through the same interfaces as an ordinary user. During the discussion of computer programs, mainly used for internet.

The purpose of the work is to develop an algorithm for analyzing and parsing the user's text for automatically generating the response of the chat bot, taking into account the topics of correspondence and morphology of the text. The algorithm's work will be based on prefix function and hash function. Also, a comparison of the developed algorithm with the existing ones will be made.

Chat-bots take a significant part in our life. Therefore, we started exploring the most used bots and found a disadvantage. The aim of the paper is the recognition of user's gender for making chat-bot's answer more human likeness. That's why we create a database with keywords (words that the bot must match), responses to them and special verb endings. But, during investigation we noticed that due to big complexity of well-known algorithm they are not as effective as we wish. According to this, we propose usage of Rabin-Karp and Knut-Pratt algorithms, due to their effectivity. In consequence of hashing we reduce a quantity of comparison which let the algorithm works faster. To add, we can find special endings with linear complexity. Results of research can be used not only for chat-bots and also for finding keywords in the text.

After analyzing existing methods, we see that the complexity of the developed algorithm for finding keywords is less than of well-know algorithm. Using the prefix function to form a bot response allows you to work with Cyrillic texts. The proposed algorithm improves work of chat bots, which determines the gender of the interlocutor. This brings the bot closer to the level of human conversation.

Надійшла до редакції 13.10.2017